# CREATE A USER IN THE DATABASE SERVER

**Syntax**

The following syntax is used to create a user in the database server.

1. **CREATE** USER [IF NOT EXISTS] account\_name IDENTIFIED **BY** 'password';

In the above syntax, the **account\_name** has two parts one is the **username**, and another is the **hostname**, which is separated by **@** symbol. Here, the username is the name of the user, and the hostname is the name of the host from which the user can connect with the database server.

1. username@hostname

The hostname is optional. If you have not given the hostname, the user can connect from any host on the server. The user account name without hostname can be written as:

1. username@%

**Step 4:** Create a new user with the following command.

1. mysql> **create** user peter@localhost identified **by** 'jtp12345';

**Step 5: Now, we will use the IF NOT EXISTS clause with the CREATE USER statement.**

1. mysql> **CREATE** USER IF NOT EXISTS adam@localhost IDENTIFIED **BY** 'jtp123456';

# Grant Privileges to the MySQL New User

MySQL server provides multiple types of privileges to a new user account. Some of the most commonly used privileges are given below:

1. **ALL PRIVILEGES:** It permits all privileges to a new user account.
2. **CREATE:** It enables the user account to create databases and tables.
3. **DROP:** It enables the user account to drop databases and tables.
4. **DELETE:** It enables the user account to delete rows from a specific table.
5. **INSERT:** It enables the user account to insert rows into a specific table.
6. **SELECT:** It enables the user account to read a database.
7. **UPDATE:** It enables the user account to update table rows.

If you want to give all privileges to a newly created user, execute the following command.

1. mysql> **GRANT** ALL **PRIVILEGES** **ON** \* . \* **TO** peter@localhost;

If you want to give specific privileges to a newly created user, execute the following command.

1. mysql> **GRANT** **CREATE**, **SELECT**, **INSERT** **ON** \* . \* **TO** peter@localhost;

Sometimes, you want to **flush** all the privileges of a user account for changes occur immediately, type the following command.

1. FLUSH **PRIVILEGES**;

If you want to see the existing privileges for the user, execute the following command.

1. mysql> SHOW GRANTS **for** username;

We can use the following query to see the list of all user in the database server: **mysql> Select user from mysql**.  
...  
**MySQL Show Users/List All Users**

1. > mysql -u root -p.
2. Enter password: \*\*\*\*\*\*\*\*\*
3. mysql> use mysql;
4. Database changed.
5. mysql> SELECT user FROM user;

MYSQL DROP USER EXAMPLE (DELETE USER

The following are the step required to delete an existing user from the [MySQL](https://www.javatpoint.com/mysql-tutorial) server database.

**Step 1:** Open the MySQL server by using the **mysql client tool**.

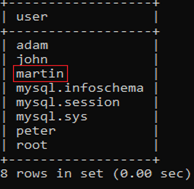
**Step 2:** Enter the password for the account and press Enter.

1. Enter **Password**: \*\*\*\*\*\*\*\*

**Step 3:** Execute the following command to show all users in the current MySQL server.

1. mysql> **select** user **from** mysql.user;

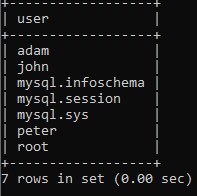
We will get the output as below:



**Step 4:** To drop a user account, you need to execute the following statement.

1. **DROP** USER martin@localhost;

Here, we are going to remove the username '**martin**' from the MySQL server. After the successful execution of the above command, you need to execute the show user statement again. You will get the following output where username martin is not present.



**Step 5:** The [DROP USER](https://www.javatpoint.com/mysql-drop-user) statement can also be used to remove more than one user accounts at once. We can drop multiple user accounts by separating account\_name with **comma** operator. To delete multiple user accounts, execute the following command.

1. **DROP** USER john@localhost, peter@localhost;

# MYSQL SHOW USERS/LIST ALL USERS

Sometimes you want to manage a database in MySQL. In that case, we need to see the list of all user's accounts in a database. Most times, we assume that there is a **SHOW USERS** command similar to SHOW DATABASES, SHOW TABLES, etc. for displaying the list of all users available in the database server. Unfortunately, MySQL database does not have a SHOW USERS command to display the list of all users in the MySQL server. We can use the following query to see the list of all user in the database server:

1. mysql> **Select** user **from** mysql.user;

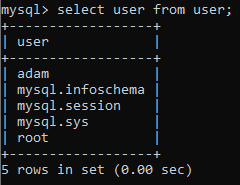
After the successful execution of the above statement, we will get the user data from the user table of the MySQL database server.

Let us see how we can use this query. First, we have to open the [MySQL](https://www.javatpoint.com/mysql-tutorial)

server by using the **mysql client tool** and log in as an administrator into the server database. Execute the following query:

1. > mysql -u root -p
2. Enter **password**: \*\*\*\*\*\*\*\*\*
3. mysql> use mysql;
4. **Database** changed
5. mysql> **SELECT** user **FROM** user;

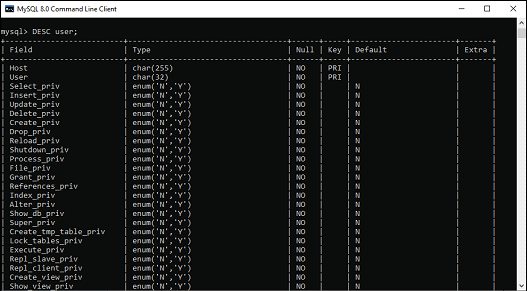
We will get the following output where we can see the **five** users in our local database:



If we want to see more information on the user table, execute the command below:

1. mysql> **DESC** user;

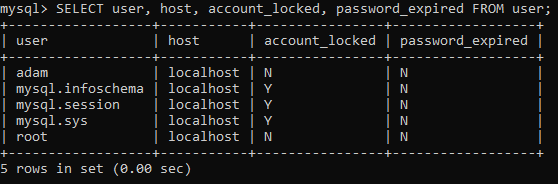
It will give the following output that lists all the available columns of the **mysql.user** database:



To get the selected information like as hostname, password expiration status, and account locking, execute the query as below:

1. mysql> **SELECT** user, host, account\_locked, password\_expired **FROM** user;

After the successful execution, it will give the following output:

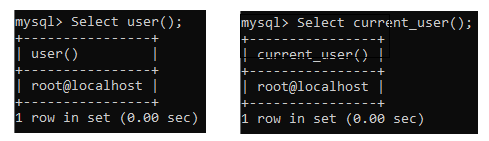


## Show Current User

We can get information of the current user by using the **user() or current\_user()** function, as shown below:

1. mysql> **Select** user();
2. or,
3. mysql> **Select** current\_user();

After executing the above command, we will get the following output:

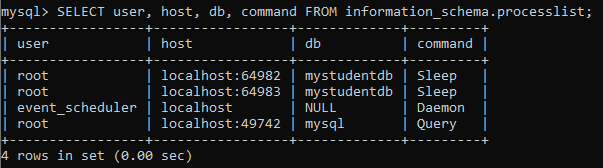


## Show Current Logged User

We can see the currently logged user in the database server by using the following query in the MySQL server:

1. mysql> **SELECT** user, host, db, command **FROM** information\_schema.processlist;

The above command gives the output, as shown below:



In this output, we can see that there are currently **four** users logged in the database, where one is executing a **Query**, and others show in **Sleep or Daemon** status.

# Change MySQL User Password

To change the password of any user account, you must have to keep this information in your mind:

* The details of the user account that you want to change.
* An application used by the user whose password you want to change. If you reset the user account password without changing an application connection string, then the application cannot connect with the database server.

[MySQL](https://www.javatpoint.com/mysql-tutorial)

allows us to change the user account password in three different ways, which are given below:

1. UPDATE Statement
2. SET PASSWORD Statement
3. ALTER USER Statement

Let us see how we can change the user account password in MySQL by using the above statement in detail:

**Change user account password using the UPDATE statement**

This statement is the first way to change the user password for updating the user table of the MySQL database. Here, you have to use the **FLUSH PRIVILEGE** statement after executing an UPDATE statement for reloading privileges from the grant table of the MySQL database.

Suppose, you want to change or update the password for a user **peter** that connects from the localhost with the password **jtp12345**, execute the [SQL](https://www.javatpoint.com/sql-tutorial)

statements as below:

1. mysql> USE mysql;
3. mysql> **UPDATE** user **SET** **password** = **PASSWORD**('jtp12345') **WHERE** user = 'peter' AND host = 'localhost';
5. mysql> FLUSH **PRIVILEGES**;

If you are using the [MySQL version](https://www.javatpoint.com/mysql-versions)

**5.7.6** or higher, the above statement will not work. It is because the MySQL user table contains the **authentication\_string** column that stores the password only. Now, the higher versions contain the authentication\_string column in the UPDATE statement, like the following statement.

1. mysql> USE mysql;
3. mysql> **UPDATE** user **SET** authentication\_string = **PASSWORD**('jtp12345') **WHERE** user = 'peter' AND host = 'localhost';
5. mysql> FLUSH **PRIVILEGES**;

**Change user account password using SET PASSWORD statement**

The SET PASSWORD statement is the second way to change the user password in the MySQL database. If you want to change the other account password, you must have the UPDATE privilege. The SET PASSWORD statement uses the user account in the **username@localhost** format.

There is no need to use the FLUSH PRIVILEGES statement for reloading privileges from the grant tables of the MySQL database. We can use the following statement to change the password of user account peter by using the SET PASSWORD statement:

1. mysql> **SET** **PASSWORD** **FOR** 'peter'@'localhost' = **PASSWORD**('jtp12345');

If you are using the MySQL version 5.7.6 or higher, the above statement deprecated and will not work in future releases. Instead, we need to use the following statement:

1. mysql> **SET** **PASSWORD** **FOR** 'peter'@'localhost' = jtp12345;

**Change user account password using ALTER USER statement**

The ALTER USER statement is the third way to change the user password in the MySQL database. MySQL uses ALTER USER statement with the IDENTIFIED BY clause for changing the password of a user account. We need to use the following syntax to change the password of a user **peter** with **jtp123**.

1. mysql> **ALTER** USER peter@localhost IDENTIFIED **BY** 'jtp123';

Sometimes, you need to reset the MySQL **root** account password. In that case, you can force to stop and restart the MySQL database server without using the grant table validation.